Nios®

Nios Development Kit, Stratix Edition

Getting Started User Guide

101 Innovation Drive
San Jose, CA 95134
(408) 544-7000
http://www.altera.com

Document Version: 1.2
Document Date: July 2003

About this User Guide

This Getting Started Guide provides initial information about how to use the Nios® Development Kit, Stratix™ Edition, including unpacking the kit, installing required software, connecting the Nios Development Board, Stratix edition to a PC, and running sample software using the reference design that is pre-installed on the development board.

Go to the following sources for more information:

- See page 20 for a list of other available Nios documentation.
- For more information about the features of this kit, visit http://www.altera.com.

Table 1 shows the user guide revision history.

Table 1. User Guide Revision History

Date	Description
July 2003	Reflects new directory structure for SOPC Builder 3.0 and Nios Development Kit version 3.1.
March 2003	Minor updates.
January 2003	Initial release - version 1.0

How to Find Information

- When using this document in PDF format, the Adobe Acrobat Find feature allows you to search the contents of the document. Click the binoculars toolbar icon in the Adobe Acrobat software to open the Find dialog box.
- Bookmarks serve as an additional table of contents.
- Thumbnail icons, which provide miniature previews of each page, provide a link to the pages.
- Numerous links, shown in green text, allow you to jump to related information.

How to Contact Altera

For the most up-to-date information about Altera® products, go to the Altera world-wide web site at http://www.altera.com.

For technical support on this product, go to http://www.altera.com/mysupport. For additional information about Altera products, consult the sources shown in Table 2.

Table 2. How to Contact Altera

Information Type	USA & Canada	All Other Locations
Product literature	http://www.altera.com	http://www.altera.com
Altera literature services	lit_req@altera.com *(1)*	lit_req@altera.com *(1)*
Non-technical customer service	(800) 767-3753	(408) 544-7000 (7:30 a.m. to 5:30 p.m. Pacific Time)
Technical support	(800) 800-EPLD (3753) (7:30 a.m. to 5:30 p.m. Pacific Time)	(408) 544-7000 *(1)* (7:30 a.m. to 5:30 p.m. Pacific Time)
	http://www.altera.com/mysupport/	http://www.altera.com/mysupport/
FTP site	ftp.altera.com	ftp.altera.com

Note:

(1) You can also contact your local Altera sales office or sales representative.

Typographic Conventions

This document uses the typographic conventions shown in Table 3.

Table 3. Conventions

Visual Cue	Meaning
Bold Type with Initial Capital Letters	Command names, dialog box titles, checkbox options, and dialog box options are shown in bold, initial capital letters. Example: **Save As** dialog box.
bold type	External timing parameters, directory names, project names, disk drive names, filenames, filename extensions, and software utility names are shown in bold type. Examples: $\mathbf{f_{MAX}}$, **\QuartusII** directory, **d:** drive, **chiptrip.gdf** file.
Bold italic type	Book titles are shown in bold italic type with initial capital letters. Example: ***1999 Device Data Book***.
Italic Type with Initial Capital Letters	Document titles are shown in italic type with initial capital letters. Example: *AN 75 (High-Speed Board Design).*
Italic type	Internal timing parameters and variables are shown in italic type. Examples: t_{PIA}, $n + 1$. Variable names are enclosed in angle brackets (< >) and shown in italic type. Example: <*file name*>, <*project name*>**.pof** file.
Initial Capital Letters	Keyboard keys and menu names are shown with initial capital letters. Examples: Delete key, the Options menu.
"Subheading Title"	References to sections within a document and titles of Quartus II Help topics are shown in quotation marks. Example: "Configuring a FLEX 10K or FLEX 8000 Device with the BitBlaster™ Download Cable."
`Courier type`	Signal and port names are shown in lowercase Courier type. Examples: `data1`, `tdi`, `input.` Active-low signals are denoted by suffix n, e.g., `resetn`. Anything that must be typed exactly as it appears is shown in Courier type. For example: `c:\quartusII\qdesigns\tutorial\chiptrip.gdf`. Also, sections of an actual file, such as a Report File, references to parts of files (e.g., the AHDL keyword `SUBDESIGN`), as well as logic function names (e.g., `TRI`) are shown in Courier.
1., 2., 3., and a., b., c.,...	Numbered steps are used in a list of items when the sequence of the items is important, such as the steps listed in a procedure.
■	Bullets are used in a list of items when the sequence of the items is not important.
✓	The checkmark indicates a procedure that consists of one step only.
☞	The hand points to information that requires special attention.
↵	The angled arrow indicates you should press the Enter key.
👣	The feet direct you to more information on a particular topic.

Notes:

Table of Contents

Getting Started

Introduction

Welcome to the Nios Development Kit, Stratix Edition! The Nios development kit is a complete embedded systems development kit for the Nios embedded processor. In addition to the full-featured Nios development board, the kit includes all the hardware and software development tools, documentation and accessories you need to begin developing Nios processor systems.

This user guide will familiarize you with the contents of the Nios development kit, and walk you through setting up your Nios development environment. In this guide, you will:

- Set up and verify correct operation of the Nios development board
- Install the development tools software
- Establish communication between the Nios development board and the host PC
- Compile C code and download to the Nios development board

When you finish this guide, you will be ready to begin designing custom Nios processor systems.

Before You Begin

Before proceeding, check the contents of the kit and verify that you received the following items:

Nios Development Board and Accessories

- Nios Development Board, Stratix Edition
- CompactFlash card (socketed in the Nios development board)
- ByteBlaster II™ parallel port download cable
- 6-foot 25-pin parallel port extension cable
- 9-pin RS-232 serial cable
- Ethernet cable
- Ethernet cross-over adapter
- LCD module
- 9-V DC power supply
- 3 regional power cables

Development Tools

Included in the kit is a CD-ROM Folder, containing the following:

- Nios Embedded Processor for Windows CD-ROM
- Nios Embedded Processor for UNIX CD-ROM
- Quartus® II Design Software for PCs CD-ROM
- Several evaluation CD-ROMs by third-party Nios development partners

Documentation

- This Getting Started Guide
- Quartus II Installation & Licensing for PCs manual
- Nios 32-Bit Programmer’s Reference Manual
- Nios 16-Bit Programmer’s Reference Manual
- Quick Start Guide for Third-Party Development Tools

The Nios Development Board

The Nios development board will be your steady companion through much of your development effort with the Nios processor. Hardware designers can use the Nios development board as a platform to prototype complex embedded systems. Software developers can use the Nios reference design pre-programmed on the development board to begin prototyping software immediately.

The Nios development board comes pre-configured with a 32-bit Nios processor hardware reference design and a software reference design stored in flash memory. Among other functions, this hardware and software reference platform includes facilities for downloading and debugging custom hardware and software designs.

☞ For complete details on the Nios development board, refer to the *Nios Development Board, Stratix Edition Data Sheet.*

Setting Up the Nios Development Board

First, you will set up the Nios development board. The Nios development board will show activity to verify that it is alive and functioning correctly.

To set up the Nios development board, follow these steps:

1. Remove the Nios development board from its anti-static shipping bag. Take care not to expose the board to electrostatic discharge (ESD) during setup or use. Also take care not to dislodge the CompactFlash card from its socket.

2. Place the board legs-down on a flat surface.

   ☞ Note the orientation of the board: The ALTERA logo on the FPGA should be right-side-up, and the Nios logo should be in the bottom-right-hand corner of the board. The rest of this document assumes this orientation when referencing locations on the Nios development board.

3. Connect the LCD module ribbon cable to connector J12. J12 is located to the right and slightly above the printed square labeled FPGA Config.

   Pin 1 on J12 must connect to pin 1 on the LCD module. Pin 1 is located on the left side of the J12 connector. The ribbon cable connector is notched on the side with pin 1. When connected correctly, the LCD module's ribbon cable will extend downward, toward you.

4. Connect the 9-V DC power-supply to the connector labeled Power (J26). Select the appropriate power cord for your geographic region. Connect one end to the 9-V DC power supply and the other end to a power outlet.

Verifying Correct Operation of the Nios Development Board

You will see activity on the board about 5 seconds after applying power. As soon as you apply power to the Nios development board, the Altera FPGA is configured with a Nios processor hardware reference design. Once the FPGA configuration is complete, the Nios processor in the FPGA wakes up, initializes itself with boot code from flash memory, and begins executing the software reference design.

Verify the following indicators of a properly functioning Nios development board:

- The LED labeled Power is on
- The LED labeled Safe is on
- The LCD displays initialization-status messages, then “Welcome to Nios!”
- The two 7-segment LEDs are active, displaying a spinning circular pattern
- The LEDs D0 – D7 are active, displaying a bouncing pattern

If you are not the first user of your Nios development board, the board may no longer contain the “default safe” image programmed in flash memory. In this case, you will not see the same indicators noted above. The LED labeled User may be on, indicating that the development board has been programmed with another designer’s user image.

☞ If you want to reprogram your board to its factory default condition, refer to the *Nios Development Board, Stratix Edition Data Sheet*.

If this is the first time you are applying power to the Nios development board and you do not see the indicators above, check all the connections and make sure that power is supplied to the board properly. For further assistance visit Altera’s online technical support web site at mysupport.altera.com.

Installing the Development Tools Software

Now that you have verified that your kit is complete and your Nios development board works correctly, you need development tools to begin developing Nios systems. This user guide makes a distinction between tools for software development and tools for hardware development. Likewise, your role as a Nios designer may be as a software developer, a hardware designer, or both. Depending on your role, you need to install certain development tools included in this kit.

Install the development tools in the following order, skipping over any tools you do not need.

The Quartus II Design Software CD-ROM

Both hardware designers and software developers need to install the Quartus II design software.

The Quartus II design software is Altera’s comprehensive environment for system-on-a-programmable-chip (SOPC) hardware design. Using the Quartus II software, designers can develop hardware design files, synthesize a netlist for the design, and output a configuration file for the target FPGA. Designers use the Quartus II software to assign I/O pin locations, apply compilation constraints (e.g. timing requirements), and perform timing analysis on the FPGA design. Quartus II software installation also includes the SOPC Builder system integration tool. Nios hardware designers use SOPC Builder to define and integrate Nios processor-based hardware systems.

The Quartus II software installation process also installs drivers necessary to use the Altera ByteBlaster II download cable. The download cable is used to download new hardware and software images to the board, and to debug embedded hardware and software.

If You are a Hardware Designer

If you are a hardware designer, perform the following steps as instructed in the *Quartus II Installation & Licensing for PCs Manual* to install the Quartus II software and the ByteBlaster II driver:

1. Install the Quartus II software. During the installation process, when you are prompted for Setup Type, select **Full Installation**.
2. To license the Quartus II software, do the following:
    a. Click the Licensing icon at the Altera web site (**http://www.altera.com**) and scroll to the Development Kit Licenses section of the Altera Licensing Center page.

    b. Click the hyperlink for your Nios development kit, and follow the on-screen instructions.
3. Specify your license file in the Quartus II software.

If You are a Software Developer

If you are a software developer, you will need the Quartus II software only to support the ByteBlaster II download cable. Perform the following steps as instructed in the *Quartus II Installation & Licensing for PCs Manual*:

✓ Install the Quartus II software. During the installation process, when you are prompted for Setup Type, select **Custom Installation** and click **Next**. The next screen will prompt you to select components. Select only the Quartus II and SOPC Builder components, and deselect all other components.

The Nios Embedded Processor CD-ROM

Both hardware designers and software developers need to install software from the Nios Embedded Processor CD-ROM. This CD-ROM contains the tools listed below.

- **The Nios CPU Component for SOPC Builder** – The Nios CPU core is a component for the SOPC Builder hardware component library.

SOPC Builder needs this component to generate the hardware and software foundation for Nios systems.

- **The GNUPro Toolkit (RedHat Inc.)**– The GNUPro Toolkit contains the basic facilities for software development such as an assembler, C/C++ compiler, linker and GDB debugger.

Hardware design and software development require different combinations of these tools. Follow the steps below to install the development tools that are right for you.

1. Insert the Nios Embedded Processor CD-ROM. If the install program does not automatically launch, browse to your CD-ROM drive and manually execute setup.exe.

2. In the Altera Nios Installer window, click **Install Nios Development Kit**.

3. Follow the installation flow until you arrive at the **Nios Development Kit Installation Type** window.

   a. If you are a hardware designer, select **Complete Nios Development Kit Installation** and click the Next button. This will install all of the Nios development tools listed above.

   ☞ The GNUPro Toolkit is required even for hardware development, because SOPC Builder performs software build operations when generating Nios processor systems.

   b. If you are strictly a software-only developer, select **Software development tools for Nios only** and click the Next button. This will install the GNUPro Toolkit.

4. Continue with the installation process.

Third-Party Software Evaluation CD-ROMs

The Nios development kit includes evaluation software from leading Nios development partners. After you familiarize yourself with the Nios processor basics, Altera invites you to enhance your Nios development environment with these tools, including a full-featured graphical debugger and JTAG-based in-system analyzer for debugging software. Refer to the *Quick-Start Guide for Third Party Development Tools* found in your kit for more information about the third-party evaluation software.

Establishing Communication Between the Nios Development Board and the Host PC

In this section, you will connect the Nios development board to your PC via an RS-232 serial cable and establish communication with the development board.

Connecting the Host PC to the Nios Development Board

To establish the physical link between the host PC and the Nios development board, follow these steps:

1. Remove power from the Nios development board, by disconnecting the power cable.

2. Connect one end of the RS-232 serial cable to a COM port on your PC.

3. Connect the other end of the serial cable to the serial connector labeled **Console** (J19), the serial port connector on the Nios development board.

4. Reapply power to the Nios development board.

Establishing Communication with the Nios Development Board

Now that the development board and host PC are connected, to establish communication between them do the following:

1. From the Windows Start Menu, select **Programs > Altera > Nios Development Kit** <*installed version*> **> Nios SDK Shell.** A new window appears and displays a Cygwin bash shell prompt.

☞ The Cygwin bash shell is part of the GNUPro Toolkit and provides a command-line interface similar to UNIX for Windows-based PCs. The bash shell is the basic Nios software development environment. More advanced graphical development environments are also available.

2. Type `nios-run -t -r` to establish a simple terminal connection with the development board

☞ Step 2 assumes the serial cable is connected to COM1 on your PC. If you are using COM2, type `nios-run -p com2 -t -r`.

☞ The **nios-run** utility, provided by Altera, establishes a simple terminal connection with the boot monitor program running on the Nios processor reference design on the board.

3. Press the Safe Config button on the Nios development board. This resets the Nios development board, and reconfigures the Stratix FPGA. When the Nios processor boots, the reference design emits a text message to the serial port.

4. Wait until the LEDs begin to blink, showing that the reference-design is running.

5. The LCD will display Network-initialization status messages. Press SW3 to abort DHCP network configuration.

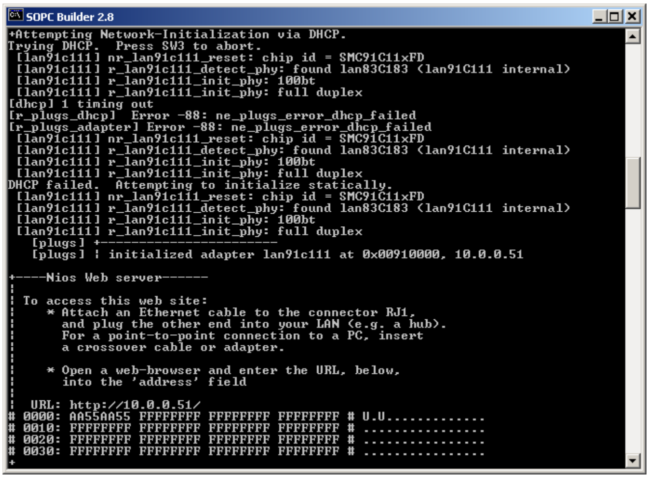
See *The Nios Development Board, Stratix Edition Data Sheet* for information about using the reference design's Ethernet features.

6. You should see new text displayed in the Nios SDK Shell window. Press the Enter key on your PC several times to provide stimulus to the reference design.

7. Wait a few moments while the reference design shuts-down the network interface. It will respond with a memory-dump and a "+" prompt, as shown in Figure 1.

   If you see text activity in the Nios SDK Shell, as shown in Figure 1, then your PC is communicating correctly with the Nios development board.

Figure 1. Nios SDK Shell Prompt



```
+Attempting Network-Initialization via DHCP.
Trying DHCP.  Press SW3 to abort.
 [lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
 [lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91C111 internal)
 [lan91c111] r_lan91c111_init_phy: 100bt
 [lan91c111] r_lan91c111_init_phy: full duplex
[dhcp] 1 timing out
[r_plugs_dhcp]  Error -88: ne_plugs_error_dhcp_failed
[r_plugs_adapter] Error -88: ne_plugs_error_dhcp_failed
 [lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
 [lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91C111 internal)
 [lan91c111] r_lan91c111_init_phy: 100bt
 [lan91c111] r_lan91c111_init_phy: full duplex
DHCP failed.  Attempting to initialize statically.
 [lan91c111] nr_lan91c111_reset: chip id = SMC91C11xFD
 [lan91c111] r_lan91c111_detect_phy: found lan83C183 (lan91C111 internal)
 [lan91c111] r_lan91c111_init_phy: 100bt
 [lan91c111] r_lan91c111_init_phy: full duplex
   [plugs] +-----------------------
   [plugs] ¦ initialized adapter lan91c111 at 0x00910000, 10.0.0.51

+----Nios Web server------
¦
¦ To access this web site:
¦     * Attach an Ethernet cable to the connector RJ1,
¦       and plug the other end into your LAN (e.g. a hub).
¦       For a point-to-point connection to a PC, insert
¦       a crossover cable or adapter.
¦
¦     * Open a web-browser and enter the URL, below,
¦       into the 'address' field
¦
¦  URL: http://10.0.0.51/
# 0000: AA55AA55 FFFFFFFF FFFFFFFF FFFFFFFF # U.U.............
# 0010: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF # ................
# 0020: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF # ................
# 0030: FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF # ................
+
```

☞ What you see is a monitor program displaying the first few lines of memory, starting at address 0x0000.

8. Press Ctrl-C to exit the terminal program and return to the bash shell.

Compiling C Code and Downloading to the Nios Development Board

In this section you will compile a sample C program and download the executable code to the Nios development board.

Compiling C Source Code

To compile C source code, do the following:

1. If you do not already have a bash shell open, then open a Nios SDK Shell. (From the Windows Start Menu, select **Programs > Altera > Nios Development Kit** <*installed version*> **> Nios SDK Shell.)**

2. Change directories to a reference design directory by typing `cd verilog/nios_stratix_1s10/standard_32/cpu_sdk/src`

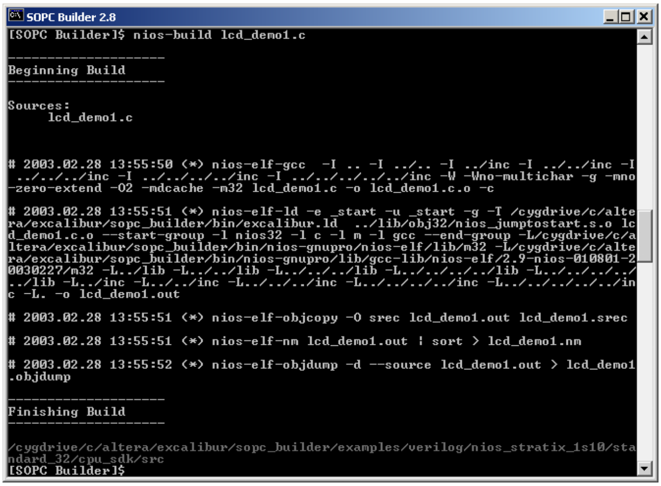
   This command places you in the default software development directory for the `standard_32` reference design.

The Nios development kit, Stratix Edition includes many example hardware and software reference designs. The `standard_32` reference design is provided in both Verilog and VHDL. This getting started user guide uses the Verilog reference design's directory for demonstration purposes.

3. To compile a simple C program that displays text on the LCD screen, at the bash prompt, type `nios-build lcd_demo1.c`.

   The nios-build utility invokes the GNU C/C++ compiler and linker, producing several intermediate files and an executable (.srec) file, as shown in Figure 2.

Figure 2. Nios-Build Message



```
[SOPC Builder]$ nios-build lcd_demo1.c

---------------------
Beginning Build
---------------------

Sources:
     lcd_demo1.c


# 2003.02.28 13:55:50 (*) nios-elf-gcc  -I .. -I ../.. -I ../inc -I ../../inc -I
 ../../../inc -I ../../../../inc -I ../../../../../inc -W -Wno-multichar -g -mno
-zero-extend -O2 -mdcache -m32 lcd_demo1.c -o lcd_demo1.c.o -c

# 2003.02.28 13:55:51 (*) nios-elf-ld -e _start -u _start -g -T /cygdrive/c/alte
ra/excalibur/sopc_builder/bin/excalibur.ld  ../lib/obj32/nios_jumptostart.s.o lc
d_demo1.c.o --start-group -l nios32 -l c -l m -l gcc --end-group -L/cygdrive/c/a
ltera/excalibur/sopc_builder/bin/nios-gnupro/nios-elf/lib/m32 -L/cygdrive/c/alte
ra/excalibur/sopc_builder/bin/nios-gnupro/lib/gcc-lib/nios-elf/2.9-nios-010801-2
0030227/m32 -L../lib -L../../lib -L../../../lib -L../../../../lib -L../../../../
../lib -L../inc -L../../inc -L../../../inc -L../../../../inc -L../../../../../in
c -L. -o lcd_demo1.out

# 2003.02.28 13:55:51 (*) nios-elf-objcopy -O srec lcd_demo1.out lcd_demo1.srec

# 2003.02.28 13:55:51 (*) nios-elf-nm lcd_demo1.out | sort > lcd_demo1.nm

# 2003.02.28 13:55:52 (*) nios-elf-objdump -d --source lcd_demo1.out > lcd_demo1
.objdump

---------------------
Finishing Build
---------------------

/cygdrive/c/altera/excalibur/sopc_builder/examples/verilog/nios_stratix_1s10/sta
ndard_32/cpu_sdk/src
[SOPC Builder]$
```

Downloading Executable Code to the Nios Development Board

To download and run the code compiled in step 3 above, do the following:

1. Type `nios-run -r lcd_demo1.srec` to download the srec file to the Nios development board. When the download is complete, program execution will begin automatically. You will see text messages appear on the LCD screen.

2. To reset the Nios CPU and return to a monitor prompt, press the CPU Reset button on the Nios development board. Press Ctrl-C to exit the terminal program and return to the bash prompt.

Taking the Next Step

Congratulations! You have completed the recommended steps to familiarize yourself with the Nios Development Kit, Stratix Edition. If you completed all of the steps above, then you have installed your Nios processor development environment, and verified that the Nios development board and tools function correctly.

Where to Go Next?

You are now ready to begin the *Nios Hardware Development Tutorial* or the *Nios Software Development Tutorial*. These tutorials introduce you to the development process for the Nios embedded processor. Each tutorial uses the Nios development board as a demonstration platform, and walks you step-by-step through the process of creating hardware and software for Nios embedded processor systems.

If you are a hardware designer, start with the *Nios Hardware Development Tutoria*l. If you will also write software for the Nios processor, continue on to the *Nios Software Development Tutorial*. If you are strictly a software-only designer, begin with the *Nios Software Development Tutorial*.

Connecting the ByteBlaster II Download Cable

In both the software and hardware development tutorials mentioned above, you will use the Altera ByteBlaster II download cable to communicate with the Stratix FPGA via the JTAG port. Before you perform either of the tutorials, connect your ByteBlaster II to the Nios development board as instructed below.

1. Remove power from the Nios development board by disconnecting the power cable.

2. Connect one end of the 25-pin parallel port extension cable to the parallel port on your PC. Connect the other end to the ByteBlaster II download cable.

3. Connect the ByteBlaster II download cable to the 10-pin header labeled ByteBlaster (J24) on the Nios development board. When connected properly, the red strip on the ByteBlaster II ribbon cable should be on the right-hand-side.

4. Re-apply power to the Nios development board.

Documentation Library

In addition to the tutorials mentioned above, Altera publishes a wealth of documentation and reference material for the Nios processor and SOPC Builder tool.

Table 1 on page 20 is a list of available documents in your Nios development kit. The table is followed by a description of each document. These documents can be accessed by choosing **F1** in SOPC Builder or by choosing **Programs** > **Altera > Nios Development Kit** <*installed version*> **> Altera Nios Documentation.** The file name of each document is found in the document's description.

☞ An Adobe® Acrobat® search feature is available to you in the **documents** directory. When you open any document in this directory, choose **Edit > Search > Query**. Enter the string of Nios/SOPC Builder-related text in the **Find Results Containing Text** dialog box. See Adobe Acrobat help for additional search information. You may need the version of Adobe Acrobat at the following link:

http://ardownload.adobe.com/pub/adobe/acrobatreader/win/5.x/5.1/AcroReader51_ENU_full.exe

Click on the title of the document in Table 1 to access the specific document on the Altera web site. Click the following link to view other available Altera literature: http://www.altera.com/literature/lit-index.html.

Table 1. Documentation Library (Part 1 of 2)

Document	Hardware Developer	Software Developer	IP Developer
Nios Development Kit Getting Started User Guide	x	x	x
Nios Development Board Document	x	x	x
Nios Hardware Development Tutorial	x		
Nios Hardware Tutorial (APEX Device)	x		
Nios Software Development Tutorial	x	x	
Nios Software Development Reference Manual		x	
GNUPro printed documentation	x	x	
FS2 System Analyzer for the Nios Processor Core Getting Started Guide (*link not available*)		x	
Plugs Ethernet Library Reference Manual		x	
Nios 32-Bit Programmer's Reference Manual		x	
Nios 16-Bit Programmer's Reference Manual		x	
SOPC Builder Data Sheet	x	x	x

Table 1. Documentation Library (Part 2 of 2)

Document	Hardware Developer	Software Developer	IP Developer
SOPC Builder PTF File Reference Manual			x
Avalon Bus Specification Reference Manual	x		
AN 184: Simultaneous Multi-Mastering with the Avalon Bus	x		x
AN 188: Custom Instructions for the Nios Embedded Processor	x	x	
AN 189: Simulating Nios Embedded Processor Designs	x		x
AN 284: Implementing Interrupt Service Routines in Nios Systems		x	x
Custom Instruction Tutorial	x	x	
Simultaneous Multi-Mastering with the Nios Processor Tutorial	x		x
Nios 3.0 CPU Data Sheet	x		x
Nios Timer Data Sheet	x	x	
Nios UART Data Sheet	x	x	
Nios SPI Data Sheet	x	x	
Nios DMA Data Sheet	x	x	
Nios PIO Data Sheet	x	x	
Legacy SDRAM Controller with Avalon Interface Data Sheet	x	x	
SDRAM Controller with Avalon Interface Data Sheet	x	x	
Active Serial Memory Interface Data Sheet (Cyclone devices only)		x	

Nios Development Kit Getting Started User Guide

This user guide (*ug_nios_<device>.pdf*) familiarizes you with the contents of the Nios development kit, walks you through setting up your Nios development environment, and provides a description of available documentation. In this guide, you will:

- Set up and verify correct operation of the Nios development board
- Install the development tools software
- Establish communication between the Nios development board and the host PC

Nios Development Board Document

This document (*ds_nios_board_<device>.pdf* or *mnl_nios_board_<device>.pdf*) describes the features and functionality of the Nios development board and provides the technical details of its components including device pin out table data.

For Nios board schematics, see the device folder in the **documents** directory.

Nios Hardware Development Tutorial

This tutorial (*tt_nios_hw_<device>.pdf*) introduces you to the Nios system module. It shows you how to use the Quartus II software to create and process your own Nios system module design that interfaces with components provided on the Nios development board.

Nios Hardware Tutorial (APEX Device)

This tutorial (*tt_nios_hw_apex_20k200e.pdf*) introduces you to the Nios system module for the Nios Development Kit, APEX Edition. This tutorial shows you how to use the Quartus II software to create and process your own Nios system module design that interfaces with components provided on the board.

Nios Software Development Tutorial

This tutorial (*tt_nios_sw.pdf*) shows the developer how to compile code, download code and debug a Nios processor system on the Nios development board.

Nios Software Development Reference Manual

This reference manual (*mnl_niossft.pdf*) provides information for programmers developing software for the Nios embedded processor. The primary focus of this document is for developers writing code in C and includes sections addressing code written in assembly.

GNUPro printed documentation

The **gnu_tools** folder in the **documents** directory contains helpful GNUPro compiler, debugger, utilities, libraries, and other tool-related documentation. There is also an on-link document at http://www.altera.com/literature/third-party/nios_gnupro.pdf

FS2 System Analyzer for the Nios Processor Core Getting Started Guide

This guide (*Nios-Getting-Started.pdf*) explains in detail the usage of the Nios OCI debug console for debugging software via the JTAG-based Nios OCI debug module.

Plugs Ethernet Library Reference Manual

This manual (*mnl_plugs.pdf*) provides a software overview introducing you to the C language "Plugs" library supporting SOPC Builder Ethernet library components. It includes a description of the supported protocols, general structure of the provided functions and data structures, and a description of the Plugs library. The Plugs library is a collection of software subroutines for SOPC Builder Ethernet components.

Nios 32-Bit Programmer's Reference Manual

This reference manual (*mnl_nios_programmers32.pdf*) contains details on the Nios processor architecture including registers, addressing modes, program-flow control, exceptions and pipeline implementation. It includes complete documentation on the Nios assembly language instruction set for a 32-bit CPU.

Nios 16-Bit Programmer's Reference Manual

This reference manual (*mnl_nios_programmers16.pdf*) contains details on the Nios processor architecture including registers, addressing modes, program-flow control, exceptions and pipeline implementation. It includes complete documentation on the Nios assembly language instruction set for a 16-bit CPU.

SOPC Builder Data Sheet

This data sheet (*ds_sopc.pdf*) contains an overview of the SOPC Builder tool which includes a description of the GUI, the module pool, the module table, arbitration priorities, the Nios CPU pages, the generation program pages, SOPC Builder library, bus protocols, interface to user-designed logic, and the SOPC Builder PTF file structure.

SOPC Builder PTF File Reference Manual

This reference manual (*mnl_sopcptf.pdf*) is for IP developers who wish to create new library components for SOPC Builder. This manual contains reference material on the internal workings of the PTF file structure and the development phases of SOPC Builder. This manual is recommended for advanced system designers with basic familiarity of the SOPC Builder tool.

Avalon Bus Specifications Reference Manual

This reference manual (*mnl_avalon_bus.pdf*) is for developers creating custom peripherals for the Avalon bus. It defines terms and concepts of SOPC designs based on the Avalon bus architecture used for connecting on-chip processors and peripherals into a system on a programmable chip (SOPC). Avalon bus signal functions and timing are defined.

AN 184: Simultaneous Multi-Mastering with the Avalon Bus

This application note (*an184.pdf*) describes the simultaneous multi-master Avalon bus with an explanation about how it differs from existing bus arbitration schemes. It includes an in-depth explanation of bus arbitration priorities and most commonly used configurations for your Nios embedded system design.

AN 188: Custom Instructions for the Nios Embedded Processor

This application note (*an188.pdf*) describes the custom instruction feature and how to implement custom instructions in Nios-based designs. It includes a design example that shows the benefits of using custom instructions.

AN 189: Simulating Nios Embedded Processor Designs

This application note (*an189.pdf*) describes the simulation flow of a simulation project and walks the user through the process of simulating a design using ModelSim® models created by SOPC Builder.

AN 284: Implementing Interrupt Service Routines in Nios Systems

This application note (*an284.pdf*) describes the basics of interrupt handling in Nios systems using the Altera-provided system software subroutines. This document uses an example ISR for an interrupt-driven UART to demonstrate these software subroutines.

Custom Instruction Tutorial

This tutorial (*tt_nios_ci.pdf*) introduces you to custom instruction using the Nios embedded processor. The tutorial guides you through the steps for implementing two example custom instructions in a Nios system module and then describes how to access these custom instructions through software.

Simultaneous Multi-Mastering with the Nios Processor Tutorial

This tutorial (*tt_nios_smm.pdf*) describes how to optimize an embedded system's performance using the simultaneous multi-master bus architecture. It describes the new features in the SOPC Builder software that allow you to customize a system bus architecture easily and shows you how to use the SOPC Builder software to define a custom bus architecture to improve the example design's performance.

Nios 3.0 CPU Data Sheet

This data sheet (*ds_nioscpu.pdf*) contains an overview of the Nios CPU core and implementation details that include instruction bus-master, data bus-master, cache memory, interrupt support, and the Nios on-chip instrumentation (OCI^{TM}) debug module. SOPC Builder PTF file assignments are described as well as software subroutines that allow communications between the Nios CPU and a host computer for connecting the Nios CPU to the Nios OCI debug module via a JTAG download cable.

Nios Timer Data Sheet

This data sheet (*ds_nios_timer.pdf*) describes the SOPC Builder Timer library component, its register map, interrupt outputs, software data structure, software subroutines, and associated SOPC Builder PTF file assignments.

Nios UART Data Sheet

This data sheet (*ds_nios_uart.pdf*) describes the SOPC Builder UART library component, its register map, interrupt outputs, software data structure, software subroutines, and associated SOPC Builder PTF file assignments.

Nios SPI Data Sheet

This data sheet (*ds_nios_spi.pdf*) describes the SOPC Builder SPI library component, its register map, software data structure, software subroutines and associated SOPC Builder PTF file assignments.

Nios DMA Data Sheet

This data sheet (*ds_nios_dma.pdf*) describes the SOPC Builder Nios DMA library component, its register map, address incrementing, interrupts, software data structure, software subroutines, and associated SOPC Builder PTF file assignments.

Nios PIO Data Sheet

This data sheet (*ds_nios_pio.pdf*) describes the SOPC Builder PIO library component, its register map, software data structure, software subroutines and associated SOPC Builder PTF file assignments.

Legacy SDRAM Controller with Avalon Interface Data Sheet

This data sheet (*ds_legacy_sdram_ctrl.pdf*) describes the SOPC Builder Legacy SDRAM Controller library component and its associated SOPC Builder PTF file assignments.

SDRAM Controller with Avalon Interface Data Sheet

This data sheet (*ds_sdram_ctrl.pdf*) describes the SOPC Builder SDRAM Controller peripheral library component and its associated PTF file assignments for interfacing with the Nios CPU.

Active Serial Memory Interface Data Sheet

This data sheet (*ds_asmi.pdf*) contains an overview of the active serial memory interface for Cyclone™ devices and a description of its functionality as an SOPC Builder library component. Included is a description of the software routines accessible through the software development kit (SDK) libraries.

For More Information

For updates and newly created documentation, see http://www.altera.com/literature/lit-nio.html or http://www.altera.com/literature/lit-sop.htm

If you need technical support, contact the Altera Applications department at http://www.altera.com/mysupport.